1. What exactly is []?

It is an empty list.

1. In a list of values stored in a variable called spam, how would you assign the value 'hello' as the third value? (Assume [2, 4, 6, 8, 10] are in spam.)

spam=[2,4,6,8]

spam[3]='hello'

spam

Let's pretend the spam includes the list ['a', 'b', 'c', 'd'] for the next three queries.

3. What is the value of spam[int(int('3' \* 2) / 11)]?=error

4. What is the value of spam[-1]?

'd'

5. What is the value of spam[:2]?= ['a', 'b']

Let's pretend bacon has the list [3.14, 'cat,' 11, 'cat,' True] for the next three questions.

6. What is the value of bacon.index('cat')?= 2

7. How does bacon.append(99) change the look of the list value in bacon?=[3.14,’cat’,’11,’cat’,’True,99]

8. How does bacon.remove('cat') change the look of the list in bacon?

[3.14,11,’True’,99]

9. What are the list concatenation and list replication operators?

List concatenation operation=

a= "megha" + "dogra"

print(a)

meghadogra

List replication=

a=’megha’\*2

meghamegha

10. What is difference between the list methods append() and insert()?

**Append adds value in end**

a=[1,2,3,4]

**a.append(5)**

a

[1, 2, 3, 4, 5]

**a.insert(0,’hello’)**

['hello', 2, 3, 4, 5]

11. What are the two methods for removing items from a list?

**1. Remove---list.remove(‘cat’)**

**2. delete by position---- del.a[-1]**

12. Describe how list values and string values are identical.

13. What's the difference between tuples and lists?

**List can be represent by square brackets- []**

**List are mutable**

**List are arranged order**

Tuple can be represented by ()

Immutable

Unordered

14. How do you type a tuple value that only contains the integer 42?

a= [42]

tuple(a)

15. How do you get a list value's tuple form? How do you get a tuple value's list form?

List=[1,2,3,4]

tuple(list)

tuple=(1,2,3,4)

list(tuple)

16. Variables that "contain" list values are not necessarily lists themselves. Instead, what do they contain?

Can be set also

17. How do you distinguish between copy.copy() and copy.deepcopy()?

sometimes you may want to have the original values unchanged and only modify the new values or vice versa. In Python, there are two ways to create copies:

Shallow Copy

Deep Copy

To make these copy work, we use the copy module.

import copy

copy.copy(x)

copy.deepcopy(x)

import copy

old\_list = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

new\_list = copy.copy(old\_list)

print("Old list:", old\_list)

print("New list:", new\_list)

import copy

x=[1,2]

copy.copy(x)